Machine Learning Engineer Nanodegree

Capstone Proposal

Aggelos Papoutsis
April 30st, 2019

Domain Background

The project is proposed by the fashion industry. The fashion industry is valued at 385.7 billion dollars[1]. Computer vision and deep learning techniques play a big role in this industry. Nowdays one of the bigger trends is models that can watch the unique preferences of one customer and launch specific recommendation on him. By this way, one company is able to engage relationships with one customer and augmented the probabilities of sales. In this direction an example and one of the bigger trends is Amazon echo which is a artificial intelligence machine that can give to anyone personal fashion advices of what to wear in terms of colors, sizes etc[2].

Problem Statement

In this project I will built a model that can act as a virtual stylish assistant. The virtual stylish assistant can help a retailer to forecast fashion trends and launch targeting marketing campaigns. The problem has different pictures of shoes, dresses, etc and the deep learning algorithm that we will use is going to classify the images into 10 categories (classes).

Datasets and Inputs

Fashion-MNIST is a dataset of Zalando's[3] article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. The original MNIST dataset contains a lot of handwritten digits. Members of the AI/ML/Data Science community love this dataset and use it as a benchmark to validate their algorithms.

[1] https://fashionunited.com/global-fashion-industry-statistics/
[2] https://www.amazon.com/Amazon-Echo-Look-Camera-Style-Assistant/dp/B0186JAEWK
[3] https://github.com/zalandoresearch/fashion-mnist

Content

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels (see above), and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

- To locate a pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27. The pixel is located on row i and column j of a 28 x 28 matrix.
- For example, pixel 31 indicates the pixel that is in the fourth column from the left, and the second row from the top, as in the ascii-diagram below.

Labels

Each training and test example is assigned to one of the following labels:

- T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot

TL;DR

- Each row is a separate image, Column 1 is the class label, Remaining columns are pixel numbers (784 total), Each value is the darkness of the pixel (1 to 255)

** **I analyze the distribution of the target class in the training set in order to see if we deal with an unbalanced set and the dataset seems to be high balanced. So there is not need for additional techniques and we can dive directly to the implementation.**

Solution Statement

So here the model is going to take each of the images and assign them into the corresponding class. We are going to use convolutional neural networks. As we deal with images we must preserve the spatial dependences between pixels. If for example we have pixels of a bag that pixels are independent on the other pixels around it. We must perform some process in order the images get into the neural network as inputs (this is the point where convolution comes in play). We will use feature detectors and then a technique called maxpooling, flatten them and fit them into the neural network.

Benchmark Model

For the benchmark model, I will use a simple 2 layer neural network and see the output. I expect that, the model will perform well but not as It can.

Evaluation Metrics

Due to the nature of our dataset (multi class classification with a balanced dataset) for evaluation metrics we can use the precision metric from the classification report. We can see all classes and with our metric we can see how many of them have been classified correctly.

Project Design

We will approach the solution with the follow steps

1. Problem statement,
2. Importing data and libraries,
3. Visualize the dataset,
4. Training the model,
5. Evaluating the model.

For the problem, it is useful to use neural networks and particular convolution neural networks. As I say before we have our training data and we will fit them into a neural network but not directly. Here convolutional neural networks begin.

1. So after image preprocessing we have a 28x28 grey image and we will run a convolutional layer,
2. We gonna use feature detectors on our images and built feature maps,
3. Use max polling in order to create smaller sets (reduced feature maps dimensionality),
4. Flatten them and finally fit them into the neural network,
5. After the feature detection process and before the pulling we gonna use relu activation in order to add non-linearity in the feature maps,
6. After the training we will see the train and test accuracy and get the prediction for the test data,
7. With plotting we will see all classes with the true and predicted values depicted,
8. We will see the true prediction using confusion matrix and classification report.